

CLAIMS:

1. A method for load balancing in a tightly-coupled multiprocessor computer system comprising the steps of:

5 placing a plurality of tasks into a centralized task queue; and

distributing the plurality of tasks in the centralized task queue to a plurality of library processors, wherein at least one task from the plurality of tasks in the centralized task queue is distributed to at least one of the
10 plurality of library processors when the library processor has at least one empty task buffer.

2. The method of claim 1, further comprising distributing the task from the plurality of tasks in the
15 centralized task queue to the one of the plurality of library processors when the one of the plurality of library processors has one or two empty task buffers, and wherein the one of the plurality of library processors has exactly two task buffers.

20 3. The method of claim 1, further comprising distributing the task from the plurality of tasks in the centralized task queue to the one of a plurality of library processors when the one of a plurality of library processors
25 has all of its task buffers empty; that is, when load of the one of a plurality of library processors is zero tasks.

4. The method of claim 1, further comprising distributing the task from the plurality of tasks in the
30 centralized task queue to the one of the plurality of library processors by the one of a plurality of library processors fetching it from the centralized task queue.

5. The method of claim 4, further comprising distributing the task from the plurality of tasks in the centralized task queue to the one of the plurality of library processors by the one of the plurality of library processors fetching it from the centralized task queue when the load of the one of a plurality of library processors is zero or one tasks.

6. The method of claim 4, further comprising distributing the task from the plurality of tasks in the centralized task queue to the one of the plurality of library processors by the one of the plurality of library processors fetching it from the centralized task queue when the load of the one of a plurality of library processors is zero tasks.

7. A method for avoiding latency in the distribution of a task from a centralized task queue to a library processor with a plurality of buffers, comprising the steps of:

preloading the task from the centralized task queue to an empty buffer of the plurality of buffers of the library processor; and

passing control to another task, ready for execution, contained in another buffer of the plurality of buffers of the library processor.

8. The method of claim 7, wherein the library processor has exactly two buffers for holding tasks.

9. A system for load balancing in a tightly-coupled multiprocessor computer system comprising

- a system kernel;
- a library task queue coupled to the kernel; and

a plurality of library processors coupled to the library task queue, wherein the system is configured for the system kernel to place tasks to be performed by the plurality of library processors into the library task queue.

5

10. The system of Claim 9, wherein at least one of the plurality of library processors further comprises a library processor kernel and one or more task buffers, and wherein the system is further configured for a task placed in the library task queue to be distributed to one of the plurality of library processors when the library processor has at least one empty task buffer.

11. The system of Claim 10, wherein the one of the plurality of library processors has exactly two task buffers.

12. The system of Claim 10, wherein the system kernel is comprised of a single processor.

20

13. The system of Claim 10, wherein the system kernel is comprised of a plurality of processors.

14. The system of Claim 10, wherein the system is further configured for the task placed in the library task queue to be distributed to the one of a plurality of library processors by the one of the plurality of library processors fetching it from the library task queue.

15. A computer program product for load balancing in a tightly-coupled multiprocessor computer system, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer code for placing a plurality of tasks into a centralized task queue; and

computer code for distributing the plurality of tasks in the centralized task queue to a plurality of library processors;

wherein a task from the plurality of tasks in the centralized task queue is distributed to one of the plurality of library processors when the library processor has at least one empty task buffer.

10

16. The computer program product of Claim 15, further comprising computer code for distributing the task from the plurality of tasks in the centralized task queue to the one of the plurality of library processors when the one of the plurality of library processors has one or two empty task buffers, and wherein the one of the plurality of library processors has exactly two task buffers.

17. The computer program product of Claim 15, further comprising computer code for distributing the task from the plurality of tasks in the centralized task queue to the one of a plurality of library processors when the one of a plurality of library processors has all of its task buffers empty; that is, when load of the one of a plurality of library processors is zero tasks.

18. The computer program product of Claim 15, further comprising computer code for distributing the task from the plurality of tasks in the centralized task queue to the one of the plurality of library processors by the one of a plurality of library processors fetching it from the centralized task queue.

19. The computer program code of Claim 18, further comprising computer code for distributing the task from the plurality of tasks in the centralized task queue to the one of the plurality of library processors by the one of the plurality of library processors fetching it from the centralized task queue when the load of the one of a plurality of library processors is zero or one tasks.

20. The computer program code of Claim 18, further comprising computer code for distributing the task from the plurality of tasks in the centralized task queue to the one of the plurality of library processors by the one of the plurality of library processors fetching it from the centralized task queue when the load of the one of a plurality of library processors is zero tasks.

21. A computer program product for avoiding latency in the distribution of a task from a centralized task queue to a library processor with a plurality of buffers, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer program code for preloading the task from the centralized task queue to an empty buffer of the plurality of buffers of the library processor; and

computer program code for passing control to another task, ready for execution, contained in another buffer of the plurality of buffers of the library processor.

22. The computer program code of Claim 21, wherein the library processor has exactly two buffers for holding tasks.